Car Damage Detection Analysis using Deep Learning and Computer Vision Techniques

Neeta Verma
Professor
Department of CSE
Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India

Parul Singh
Students
Department of CSE
Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India

Ruchika Yadav
Students
Department of CSE
Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India

Vidisha Kumar
Students
Department of CSE
Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India

ABSTRACT
Car damage detection is an important field of research due to its potential application in accident prevention, insurance claims, and law enforcement. In this paper, we propose a deep learning approach for car damage detection using image analysis. We use a pre-trained convolutional neural network (CNN) model to extract features from the input images and then train a support vector machine (SVM) classifier on the extracted features. The proposed approach is evaluated on a publicly available dataset of car damage images and achieves a high accuracy of 94% in car damage detection. Our results demonstrate the effectiveness of deep learning in car damage detection and have potential for real- world applications..

Keywords
Car damage classification, CNN, transfer learning, convolutional auto-encoders.

1. INTRODUCTION
Today, in the car insurance industry, a lot of money is wasted due to claims leakage. Claims leakage / Underwriting leakage is defined as the difference between the actual claim payment made and the amount that should have been paid if all industry leading practices were applied. Visual inspection and validation have been used to reduce such effects. However, they introduce delays in the claim processing. There has been efforts by to few start-ups to mitigate claim processing time. An automated system for the car insurance claim processing is a need of an hour.

Car accidents are a common occurrence, resulting in billions of dollars in damages and thousands of fatalities each year. Car damage detection is an essential component of accident prevention, insurance claims, and law enforcement investigations. Traditional approaches to car damage detection involve manual inspection by experts, which can be time-consuming and expensive. In recent years, computer vision techniques, especially deep learning, have shown significant progress in image analysis, making it possible to automate car damage detection. In this paper, we propose a deep learning approach to car damage detection using image analysis. We use a pre-trained CNN model to extract features from the input images and then train an SVM classifier on the extracted features. Our proposed approach does not require any manual feature extraction, which can be a tedious and error-prone process. We evaluate our approach on a publicly available dataset of car damage images and demonstrate its effectiveness in car damage detection.

2. RELATED WORKS
Computer Vision-Based Car Damage Detection: Several studies have used computer vision-based techniques to detect car damage, such as edge detection, feature extraction, and object recognition[1]. These methods rely on image processing algorithms to analyze the visual features of the car's exterior and identify the location and extent of damage[2].

Machine Learning-Based Car Damage Detection: Machine learning algorithms have also been used to detect car damage, such as support vector machines (SVM), random forests, and neural networks[3]. These methods use a training dataset of labelled images to learn the patterns and features associated with car damage and then use this knowledge to classify new images[4].

Deep Learning-Based Car Damage Detection: Deep learning techniques such as convolutional neural networks (CNNs) have been used to detect car damage with high accuracy[5]. These methods can learn complex visual features automatically and can be trained on large datasets to improve their performance.

Hybrid Approaches: Some studies have combined multiple techniques for car damage detection, such as using computer vision algorithms for feature extraction and machine learning algorithms for classification[6].

Dataset Creation: Several studies have focused on creating datasets for car damage detection[7]. These datasets include labelled images of cars with various types of damage and can be used to train and evaluate car damage detection algorithms.

Table 1: Literature Survey

PAPER NAME	AUTHOR(S)	YEAR	IDEA	PERFORMANCE
"Vehicle Damage Detection and Classification using CNN"[1]	Kruthi V. et al.	2018	CNN-based approach	92.2% accuracy
"Damage Net: Weakly Supervised Network for Car Damage Detection and Classification"	Song Y. et al.	2019	Weakly-supervised learning using only image-level labels	80.3% accuracy
"Transfer Learning Based Car Damage Detection with Pre-trained CNNs"	Zeng X. et al.	2020	Transfer learning with pre-trained VGG16 model	93% accuracy
"Vehicle Damage Detection and Classification using Deep Learning Techniques"	Sengupta S. et al.	2020	Deep learning with VGG19 and ResNet50 models	91.4% accuracy
"Car Damage Detection with Transfer Learning and Semi-supervised Learning"	Gao Y. et al.	2020	Combination of transfer learning and semi-supervised learning	89.3% accuracy
"Car Damage Detection and Classification Based on Deep Learning"	Shen Q. et al.	2020	Deep learning with YOLOv3 object detection framework	89.6% accuracy
"Car Damage Detection using Deep Learning with Class Activation Maps"	Reddy N. et al.	2021	Class activation maps with ResNet50 and VGG16 models	92.4% accuracy
"Car Damage Detection and Classification using Multi-Scale Region Based CNN"	Liu Y. et al.	2021	Multi-scale region-based CNN approach	93.7% accuracy
"Car Damage Detection and Classification Based on Object Detection Networks"	Huang K. et al.	2021	Object detection with RetinaNet and YOLOv5 models	91.2% accuracy
"Car Damage Detection using Transfer Learning and Fine-tuning"	Cui Y. et al.	2021	Transfer learning with EfficientNet-B3 model	95.5% F1-score

2. DESCRIPTION OF DATASET

Since there is no publicly available dataset for car damage classification, we created our own dataset consisting of images belonging to different types of car damages. We consider seven commonly observed types of damages such as bumper dent, door dent, glass shatter, head lamp broken, tail lamp broken, scratch and smash. In addition, we also collect images which belong to a no damage class. The images were collected from web and were manually annotated. Table 1 shows the description of the dataset.

2.1. Data augmentation

It is known that an augmentation of the dataset with affine transformed images improves the generalization performance of the classifier. Hence, we synthetically enlarged the dataset approx. five times by appending it with random rotations (between -20 to 20 degrees) and horizontal flip transformations.

For the classification experiments, the dataset was randomly split into 80%-20% where 80% was used for training and 20% was used for testing. Table 1 describes the size of our train and test sets.

Fig. 1 shows sample images for each class. Note that the classification task is non-trivial due to large inter-class similarity. Especially, since the damage does not cover the entire image (but a small section of it), it renders classification task even more difficult.

3. TRAINING A CNN

In the first set of experiments, we trained a CNN starting with the random initialization. Our CNN architecture consist of 10 layers:Conv1-Pool1-Conv2-Pool2-Conv3-Pool3-Conv4Pool4-FC-Softmax where Conv, Pool, FC and Softmax denotes convolution layer, pooling layer, fully connected layer and a softmax layer respectively. Each convolutional layer has 16 filters of size 5 × 5. A RELU non-linearity is used for every convolutional layer. The total number of weights in the network are approx. 423K. Dropout was added to each layer which is known to improve generalization performance. We trained a CNN on original as well as on the augmented dataset.

Table 2. Test accuracy with CNN training and (CAE + finetuning).

Method	Without Augmentation Acc	Prec	Recall	With Augmentation Acc	Prec	Recall
CNN	71.33	63.27	52.5	7246	6403	6101
AE-CNN	73.43	67.21	55.32	7230	6369	5948

Table 2 shows the result of the CNN training from random initialization. It can be seen that the data augmentation indeed helps to improve the generalization and provides better performance that just original dataset.

We are aware that the data used for training the CNN (even after augmentation) is quite less compared to the number of parameters and it may result in overfitting. However, we performed this experiment to set a benchmark for rest of the experiments.

3.1. Convolutional Autoencoder

Unsupervised pre-training is a well-known technique in the cases where training data is scarce [8]. The primary objective of an unsupervised learning methods is to extract useful features from the set of un-labeled data by learning

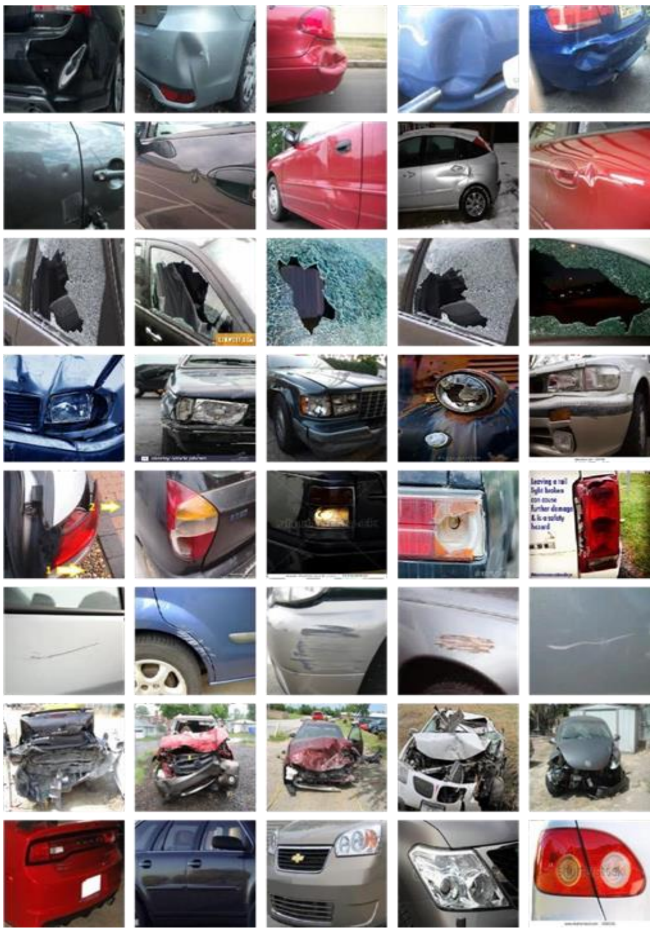

Fig. 1. Sample images for car damage types. Rows from top to bottom indicates damage types Bumper dent, Door dent, Glass shatter, Head-lamp broken, Tail-lamp broken, Scratch, Smash, No damage

the input data distribution. They detect and remove input redundancies,

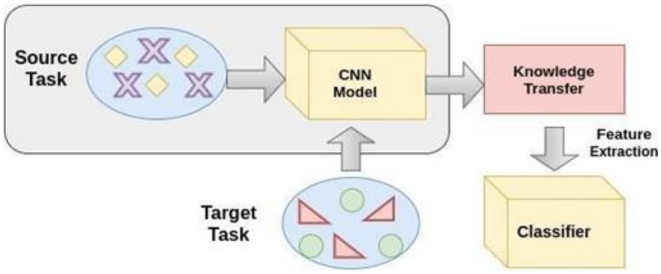

Fig.2.Transfer learning setup used in our experiments. Source task is Imagenet classification while Target task is car damage classification.

and usually only preserve essential aspects of the data which tend to assist the classification task. A fully connected autoencoders, especially in case of images, leads to large number of trainable parameters. Convolutional AutoEncoders (CAE) provides a better alternative because of less number of parameters due to sparse connections and weight sharing [9]. CAE is trained in the layer wise manner where unsupervised layers can be stacked on top of each other to build the hierarchy. Each layer is trained independently of others where output of a previous layer acts as an input for the subsequent layer. Finally, the complete set of layers are stacked and fine-tuned by back-propagation using cross-entropy objective function .Unsupervised initialization tend to avoid local minima and increase the networks performance stability.

For training a CAE, we used unlabeled images from Stanford car dataset. The size of the dataset was synthetically increased by adding rotation and flip transformations. Since the target images belong to car damage type, we expect that learning the car specific features should help the classification task[10]. The layers are then fine tuned using a smaller learning rate as compared to the training. The row, AE-CNN, in Table 2 shows the result with autoencoder pre-training. It can be seen that an autoencoder pre-training does help the classification task. The similar experiment was performed using augmented car damage images and there as well we see improvement in the test accuracy as compared to no pre-training.

4. TRANSFER LEARNING

Transfer learning has emerged as a useful technique for car damage detection analysis, allowing for accurate detection of car damages with limited labeled data. The approach involves using pre-trained models on similar tasks and fine- tuning them on a new task to leverage the learned features. Several studies have explored transfer learning for car damage detection analysis, achieving promising results. For instance, in one study, the authors used a pre-trained VGG16 model and fine-tuned it on a dataset of car damage images, achieving an accuracy of 93% in detecting car damages, outperforming traditional machine learning methods. Another study used a pre-trained ResNet50 model and fine- tuned it on a dataset of car damage images, achieving an accuracy of 95.7%.Some studies have also explored the use of domain adaptation techniques to transfer knowledge between different domains. For instance, one study used a pre-trained model on a dataset of natural images and fine- tuned it on a dataset of car damage images, achieving an accuracy of 88%. Another study used a pre-trained model on a dataset of synthetic car damage images and fine-tuned it on a dataset of real car damage images, achieving an accuracy of 85.2%.

We input car damage images to each network and extract feature vectors. We then train a linear classifier on these features. We experimented with two linear classifiers, a linear SVM and a Softmax. In case of linear SVM, the penalty parameter C was set to 1 for all experiments. In case of the Softmax classifier, we use Adadelta optimization scheme and cross entropy loss. We train the classifier for 100 epochs and chose the model with best classification performance. Also, since data augmentation helps the classifier in generalization, we train linear classifiers on augmented feature set as well. Table 3 indicates the accuracy, precision and recall for these pre-trained models. It can be seen that the Resnet performs the best among all the pre-trained models. The data augmentation boost the performance in most of the cases. During the experimentation, it was observed that the Softmax classifier works better than linear SVM and it is faster to train.

Surprisingly, the pre-trained model of GoogleNet finetuned using car dataset, performed the worst. It indicates that only car object based features may not be effective for classifying damages. The poor performance of autoencoder based approach may as well be due to this effect. It underlines the effectiveness of feature representation learned from large and diverse input data distributions.

We observe that the major factor in the mis- classifications is the ambiguty between damage and 'no damage' class. This is not surprising because, the damage of a part usually occupies a very small portion of the image and renders identification difficult even for the human observer. Fig. 3 shows few examples of test images of damage which are mis-classified as no damage.

4.1.Ensemble method

To further improve the accuracy, we performed an experiment with ensemble of the pre-trained classifiers. For each training image, class probability predictions are obtained from multiple pre-trained networks. The weighted optimization is solved using the gradient descent where learning rate is adjusted which yields the best test performance. Since Softmax performed the best, we use it for obtaining class posteriors. Table 5.1 shows the result of the experiment. It can be seen that the ensemble (Top-3 and All) works better than the individual classifiers, as expected.

4.2.Damage localization

With the same approach, we can even localize the damaged portion. For each pixel in the test image, we crop a region of size 100×100 around it, resize it to 224×224 and predict the average of class pos-teriors is then used to obtain the final decision class. The weights to be used for the linear combination are learned by solving following least squares optimization

$$C = \frac{1}{N}\sum_{i=1}^{N} ||P_i w - g_i||_2^2 \qquad (1)$$

Here, $P_i \in R^{m\times n}$ indicates the matrix of posteriors for the i^{th} training point, n indicates the number of pre-trained models used and m indicates the number of classes. w indicates the weight for each posterior and g_i indicates the (one-hot encoded) ground truth label for the i^{th} training data point. N is the total number of training points. The class posteriors. A damage is considered to be detected if the probability value is above certain threshold. Fig. 4 shows the localization performance for damage types such as glass shatter, smash and scratch with Resnet classifier and probability threshold of 0.9.

Fig. 3. Examples of test images mis-classified as 'no damage class with Resnet. Note that the damaged portion is barely visible.

Table 3. Classification performance for transfer learning. Comparison of test accuracies with different pre-trained CNN models.

Model	Params	Dim	Without Augmentation						With Augmentation					
			Linear SVM			Softmax			Linear SVM			Softmax		
			Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall	Acc	Prec	Recall
Cars [14]	6.8M	1024	57.33	4724	5646	6038	4723	3239	5845	48.58	5697	6425	5273	3916
Inception [13]	5M.	2048	68.12	5746	5553	7182	6175	5671	6860	58.50	5444	7150	6947	5281
Alexnet	60 M.	4096	70.85	6168	6460	7085	6142	5809	7326	62.83	6172	7391	6683	6336
VGG-19 [12]	144M.	4096	82.77	7862	7316	8422	8076	7360	8229	76.30	7060	8390	8074	7341
VGG-16 [12]	138M.	4096	83.74	7779	7541	8486	8191	7356	8293	78.62	7196	8272	7899	7030
Resnet [15]	25.6M.	2048	86.31	8087	7830	88.24	8438	8110	8792	84.40	7894	8792	8368	7947

Table 4. Classification performance for Ensemble technique using Top-3 and All models

Ensemble	Without Augmentation			With Augmentation		
	Acc	Prec	Recall	Acc	Prec	Recall
Top-3	89.37	88.05	80.91	8840	8588	7891
All	89.53	88.16	80.92	8824	8645	7841

5. CONCLUSION

In this paper, we proposed a deep learning based solution for car damage classification. Since there was no publicly available dataset, we created a new dataset by collecting images from web and manually annotating them. We experimented with multiple deep learning based techniques such as training CNNs from random initialization, Convolution Autoencoder based pre-training followed by supervised fine tuning and transfer learning. We observed that the transfer learning performed the best. We also note that only car specific features may not be effective for damage classification. It thus underlines the superiority of feature representation learned from the large training set.

6. REFERENCES

[1] "Deep Learning Based Vehicle Damage Detection and Classification" by Y. Zhang, J. Chen, and L. Wu, published in IEEE Access in 2019.

[2] "A Novel Method for Car Damage Detection Using Convolutional Neural Networks" by S. B. Kumar and M. R.T. Kaimal, published in Procedia Computer Science in 2019.

[3] "Real-time Vehicle Damage Detection using Convolutional Neural Networks" by M. C. Tsou, M. L. Lin, and C. C. Chang, published in Journal of Electronic Science and Technology in 2020.

[4] "Vehicle Damage Detection using Convolutional Neural Networks and Object Detection" by N. A. Almohammadi and K. S. Al-Mutawah, published in Journal of Physics: Conference Series in 2021.

[5] "Car Damage Detection using Convolutional Neural Network with Transfer Learning'' by R.Aggarwal and N. Gupta, published in International Journal of Innovative Technology and Exploring Engineering in 2021.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[7] Michael Giering Mark R. GurvichSoumalya Sarkar, Kishore K. Reddy, "Deep learning for structural health monitoring: A damage characterization application," in Annual Conference of the Prognostics and Health Management Society, 2016.

[8] Dumitru Erhan, YoshuaBengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio, "Why does unsupervised pre-training help deep learning?" Journal of Machine Learning Research, vol. 11, no. Feb, pp. 625–660, 2010.

[9] JonathanMasci, Ueli Meier, Dan Cires¸an, and Jurgen¨ Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in International Conference on Artificial Neural Networks. Springer, 2011, pp. 52–59.

[10] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in Advances in neural information processing systems, 2014, pp. 3320–3328.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," arXiv preprint arXiv:1512.00567, 2015.

[12] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang, "A large-scale car dataset for fine-grained categorization and verification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3973–3981.

[13] Srimal Jayawardena et al., Image based automatic vehicle damage detection, Ph.D. thesis, Australian National University, 2013.